



[Maxim/Dallas](#) > [App Notes](#) > [1-WIRE® DEVICES](#)

Keywords: software authorization, copy protection, piracy, DS1425, DS1427, software distribution management, software licensing, iButton, iButtons

Feb 16, 1999

APPLICATION NOTE 96

Dallas Buttons Ease Software Distribution Management

As software piracy increases, the addition of copy protection is also becoming more important. These issues are in addition to the multiple platform support, evaluations, and, in some cases, already complex license management issues that already exist. Dallas Semiconductor offers a solution called iButtons® that provide software developers with a common foundation to base copy protection, complex license management, and distribution policies.

Introduction

Software vendors are increasingly being asked by their customers to support a wider variety of distribution and license. As software piracy increases, the addition of copy protection is also becoming more important. These issues are in addition to the multiple platform support, evaluations, and, in some cases, already complex license management issues that already exist. Software distribution and license management is now very difficult, in large part because all of the pieces don't often integrate together easily.

Dallas Semiconductor offers a solution called Buttons that provide software developers with a common foundation to base copy protection, complex license management, and distribution policies.

Introducing Buttons into software distribution schemes allows coordination of a diverse set of management policies, and minimizes the cost of maintaining the policies. Savings are achieved by utilizing the advanced features of the Button architecture to reduce (and in some cases eliminate) the cost of materials and administrative overhead associated with maintaining a strategy.

Using Buttons for Copy Protection

Traditionally, there has been software and hardware approaches to protecting. Software only approaches typically use (unprotected) computer memory and a clock as base resources. This approach represents the lowest cost for protection but often is easily defeated.

The concept of offering a family code or serial number in an external hardware device has traditionally characterized the hardware approach. These devices are called dongles. Although dongles are still being used, they too are rather easily defeated.

Buttons represent a combination of the best of both traditional approaches with additional, and far more advanced capabilities.

Like the software approach, Buttons use memory and clock resources to base a protection scheme. The difference is these resources are inside the Button and are tamperproof. Button memory is password protected, and the Button clock can be permanently locked. In addition, each Button contains a unique and unalterable serial number, which has great utility when transferring licenses or creating different password schemes for each software copy sold, a level of granularity traditional approaches cannot achieve.

Button based approaches focus more on execution control rather than copy protection. If the presence and

validity of a Button determines execution, then the number of unauthorized copies of an application that are made is no longer an issue, because execution is based on the number of Buttons distributed, not the number of copies made.

The change in focus from copy protection to execution control greatly simplifies the ability to effectively protect against unauthorized use, but also provides the mechanism whereby many types of licenses can be supported (such as fixed, floating, and temporary licenses). By choosing the Button with the right functionality, varying license policies can be quickly supported.

Lower Protection Costs by Using One Button to Protect Multiple Applications

Software vendors who sell multiple applications or a base product with options can lower their overall protection costs by amortizing them across more than one revenue stream. The DS1425 Button is typically used in this application, which can provide execution control for multiple applications. It is a nonvolatile RAM device that is password protected. Software vendors can read and write tables of their own security directly into the Button. Setting the RAM password prevents unauthorized user modifications.

As a result, independent authentication for each application is achieved while the physical storage of the security resides in one Button.

Traditionally, one dongle per application is required. A security code is assigned to the software vendor by the security device provider. This approach does not give the software developer control of the security, does not allow for changes, or more than one security code. This lack of control limits the security barrier and eliminates economic gains over multiple applications. The developer has a way of storing more than one security code. Traditional approaches, therefore, force a cost floor on the scheme.

For example, suppose Company A sells three software applications as a base product (for \$100) with two add-on options (for \$50 each). Using Buttons, many security codes can be loaded into one DS1425. However, to simplify the example, assume Company A loads only three. The protection costs for all three products is \$25 (\$19 for a Button Holder and \$6 for a Button).

If the \$25 protection cost is amortized over the revenue streams of the three products, such that the base product absorbs one half the cost (\$12.50), and the options each absorb a quarter of the cost (\$6.25), the impact on the revenue streams of all three products is 12.5%. The cost impact is then lowered by spreading it equitably across the product line. Note that it did not cost Company A any additional actual dollars to do this since one Button is needed to protect one or all of the products. The decision to protect the base product using Buttons yielded free protection for the additional two options.

If Company A offers more options in the future, they can add more codes to the one DS1425 Button and further stretch their protection costs over even more revenue streams.

The traditional approach requires one device per application, or \$75, to protect the base product and the options (assuming a \$25 per device cost). This represents 37.5% of each product's revenue stream. Since this is excessive, the more likely scenario is to use the same device to protect all three applications, reducing the cost to \$25.

However, this forces the software vendor to rely on the integrity of the exact same security code (assigned by the security device provider) for the base product and the options. While this costs the same as a Button approach, it compromises security, because the same code is used for all products. Buttons offer higher security for no additional cost.

One Button Per Option Still Saves Money

In situations where only a single Button per option is more logistically practical, multiple Buttons can connect together (multidrop). This leverages the electronics from one holder across all Buttons, saving the cost of purchasing additional holders for the other applications.

The multidrop capability of the Buttons is made possible because each Button contains a unique 64 bit identification number that is unalterable and guaranteed to be unique. Software routines can identify strings of Buttons connected together using the 1-Wire™ protocol by their unique identification numbers. A benefit is that users can insert the Buttons into Button ports in any order.

Dallas Semiconductor offers low-cost extension panels that connect to a Button Holder, extending the number of Button ports up to 24. Adding an extension, on a per Button port basis is less expensive than adding additional Button Holders.

Suppose Company A used an individual Button to protect each of their three products. Three Button ports would then be required to accommodate Buttons for a base product and two options. The protection costs would grow from \$ 25 to \$50 (adding \$12 for two more Buttons and \$13 for a 4-port extension panel), when compared with the earlier example.

However, the additional \$25 for the panel can be amortized over the two options, adding only \$12.50 to each option's cost structure. This is still less expensive than adding another Button Holder, and the combined cost of the amortized panel and Button ($\$12.50 + \$6.00 = \$18.50$) is still less expensive than the \$75 traditional approach.

Lower Material and Overhead Costs for Version Updates

Another important feature of the Button scheme is that they can be removed from their holders and replaced.

Holders only need to be purchased initially (\$19), lowering the cost of protection scheme changes to the cost of the Button (\$6). This advantage is particularly important when issuing version updates.

With the traditional dongle method of protection, no savings are gained, because the electronics used for the security scheme is tightly coupled to the dongle and sold as one unit. This forces software vendors to absorb the material costs for the holders in addition to the new electronics each time a protection scheme change is desired.

The following chart shows how the two approaches compare:

	Holder Cost	Button Cost	Total Cost	Version Update Cost	Overall Costs	Cost Per Version
Traditional Dongles	\$25	N/A	\$25	\$25	\$50	\$25
Using Buttons	\$19	\$6	\$25	\$6	\$31	\$15.50

The Button approach lowers the impact on the cost of the protection scheme from \$25 per version to \$15.50 (a 38% savings). This is particularly important if the software vendor cannot command the same resale price for version updates than for the initial sale.

Using the example from before, if Company A were to just protect the base product (which resells for \$100), but could only command a \$50 resale price for a version update, the Button approach would allow Company A to perhaps absorb the initial cost (\$25) into the \$100 revenue stream, and realize the savings during the version updates, because only a Button is required (\$6) to make a protection change.

Note also that it is practical to eliminate the update cost entirely. This can be achieved by either recovering the

originally issued Buttons and recycling them or by using the installation process for the version update to change the protection. It is possible to create routines for the installation process that modifies the Buttons. The overhead is relatively low and would happen transparently to the user.

Recycling or reprogramming the Buttons lowers the cost per version update (relative to our example) from \$15.50 to \$12.50. If additional updates are released, the approach allows for further amortization and even lower per version protection costs.

Up until now, this section has concentrated on material cost for version updates. However, there is also a hidden savings when using Buttons. Programming them can be automated, reducing the overhead costs associated with their distribution.

Company A can have many different combinations of evaluations and sales taking place simultaneously. An easy and low overhead method of managing the process is to create a database which stores Button ID numbers, ties them to sales order or work order numbers (which identifies a specific evaluator or customer) using a relational database, and also records the type of Button distributed (a DS1427 for evaluations, a DS1425 for permanent licenses).

Shipping or manufacturing employees read the Button ID, store it into the database, identify the option (from the work order), and store the appropriate security codes in the Button. Programming the Buttons' password to protect the security codes completes the process. Software routines can be developed to automate this activity, so that the physical action to complete all the tasks is a momentary touch of the Button to a Button Reader, affixed to a PC in the shipping department.

The overhead cost for programming and distributing Buttons has now been reduced to a mechanical, repetitive process. And since there is a database of good information now available, marketing departments can analyze usage of the product, who the evaluators are, and how long they have been evaluating the product(s). Applications engineers can use the database to help troubleshoot problems. Accounting departments can use the database to ensure each customer has been correctly billed and has paid for the software they purchased.

While direct monetary savings cannot be placed on the value of the database, it certainly offers efficiency gains beyond the original security intentions of using Buttons. The technology helps to make the business run more efficiently.

Implementing a Low Overhead Rental License Strategy

Suppose that Company A's customer wanted to temporarily license an option product for 30 days to cover a peak engineering demand and accelerate a product's time to market. Traditionally, protection devices would have little or no value. Company A would then issue an ad hoc temporary license to enable the use of the software.

However, using the Button scheme, shipping a DS1427 Time Button and pre-programming its unalterable realtime clock would allow Company A to issue a temporary license, retain audit control, and either disable the license or convert it to a permanent one by issuing a DS1425. All of the benefits discussed earlier apply. Additional savings are realized because the overhead to convert the evaluator into a customer has been minimized.

Rentals then become a natural part of the licensing scheme when using Buttons. In addition, security is enhanced, because the time source to base billing on is unalterable.

Lowering the Cost of Maintaining Pre-Sales Product Evaluations

Rental schemes can also be used as a powerful tool in the pre-sales process. Using the shipping database

concept discussed earlier, and issuing evaluations with time expiring Buttons, creates an audit trail database which can be used to follow up on inquiries.

Demonstration-only versions of the software are no longer required (eliminating the cost of packaging special versions of the products and the cost of engineering). The time expiring Button provides the limited execution control for the actual product, the very objective of the demonstration disks. Shipping a full set of features allows the evaluator to try the product and experience its full capability, increasing the chances of a sale.

Suppose Company A decided to market its products by mail. Using Buttons, one cost effective approach would be to advertise 30-day trials for the product(s) given the reader calls an 800 number. Sales personnel could answer the 800 phone line and enter the evaluator's information into the database. Since the programming of the Buttons is automated, the sales person registers the evaluation quickly by choosing a blank DS1427 Button and touching it to a reader (initializing the clock and registering the Button ID).

The software is shipped and a record exists in a database that can be accessed by a sales or telemarketing group. When the evaluation is complete, conversion to a permanent license is simply a matter of issuing a DS1425 Button, perhaps also a task for the sales person. Registering the DS1425 Button ID could also automatically trigger an invoice. The entire process is complete with virtually no paperwork.

This method of conversion also eliminates the cost of shipping new media, commonly incurred when distributing demo disks and production software. The customer already has the software, they just need a new Button.

Lower Protection Costs in Multiple Vendor Solutions

It is quite possible for a software vendor that offers a base package to establish a service business that coordinates and distributes the protection keys for all the third party option providers. This business would allow all parties to lower their cost of protection.

The concept is similar to the one discussed earlier, except that a fee is charged by the company willing to take on the service, for rental space inside a Button, or for deploying a separate Button. The base vendor could then amortize the cost of the Button Holders and extension panels over the number of third party vendors participating, offsetting the actual costs with revenues accrued from the service.

Suppose Company A provided just a base product. Companies B and C each supply options to Company A's base products. If Company A agrees to purchase the Button Holder (\$19), a four port Button extension panel (\$13), and a Button for their own protection (\$6), Company A's investment is \$38 (or 38% of its \$100 revenue stream).

If Company A then distributes Buttons for the option providers, and charges \$12.50 for the service, the sales of Company B and C products will result in \$25 of revenue. If 25% of the service revenue is used to pay for the overhead and cost to manage the service, \$18.75 can be applied to the original protection cost of \$38, lowering it from \$25 (\$19 Button holder plus \$6 Button) to \$19.25. Companies B and C receive a 50% cost savings, compared to the \$25 cost of independent protection. As a result, all parties benefit and the customer receives an easy to manage multi-vendor protection scheme.

Compatibility with License Servers

In some instances, complex licensing requires the use of dedicated server-based products to administer the software. This is often the case when considering floating license policies for network applications.

Dallas Semiconductor has partnered with many of the leading suppliers of license server products to ensure compatibility between the two forms of technology. This means Dallas Button products will operate seamlessly with the license server. Contact Dallas Semiconductor for a current list of compatible vendors.

"Button Ready" Computers

Dallas Semiconductor is the only software management vendor that has successfully integrated the security electronics into a distributed computer. The Button Holders have been absorbed and their cost can be eliminated by selecting a vendor that offers a "Button Ready" computer. Call Dallas Semiconductor for a current list of vendors.

Button Ready computers means that Dallas' technology is destined to become the defacto standard for distributed computing security hardware. Choosing Dallas now ensures compatibility in the future.

Application Note 96: <http://www.maxim-ic.com/an96>

More Information

For technical questions and support: <http://www.maxim-ic.com/support>

For samples: <http://www.maxim-ic.com/samples>

Other questions and comments: <http://www.maxim-ic.com/contact>

Related Parts

DS1410E: [QuickView](#) -- [Full \(PDF\) Data Sheet](#)

DS1411: [QuickView](#) -- [Full \(PDF\) Data Sheet](#)

DS1413: [QuickView](#) -- [Full \(PDF\) Data Sheet](#)

DS1425: [QuickView](#) -- [Full \(PDF\) Data Sheet](#)

DS1427: [QuickView](#) -- [Full \(PDF\) Data Sheet](#)

DS1963S: [QuickView](#) -- [Free Samples](#)

DS1991: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

DS1994: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

AN96, AN 96, APP96, Appnote96, Appnote 96

Copyright © 2005 by Maxim Integrated Products

Additional legal notices: <http://www.maxim-ic.com/legal>